

Service Orientation Promotes Business Innovation

'There is nothing permanent except change', said Heraclites in the 6th Century BC, and how right he was! The relatively prosperous period leading up to the latest economic crisis was marked by a spate of mergers and acquisitions that resulted in constant internal change for many business and technical organisations. The financial crisis then brought a welter of changes in the external environment of an organisation so that, to survive, it had to react to those changes immediately.

It seems that in both good and bad times Business innovation is the key to the organisation's wellbeing. Occasionally, an innovation may be small - for example, the adoption of minor changes to the way something is done. At other times the change may be large - expressed on the level of company-wide strategic initiatives. To win in the modern market, however, it doesn't only matter what you change; equally important is *how you* carry out your changes, how quickly you can adapt to external change and modify your business functions and operations accordingly.

The concept of service orientation addresses adaptability and flexibility, i.e. promotes all necessary mechanisms for innovation. We have known this for years, in spite of some obfuscation regarding the technology. So why don't we use it to its full capacity? What stops organisations from adopting it, and thus impedes their efforts to adapt to market changes?

This chapter discusses the detachment between management responsibilities and corporate efficiency, examines the view of the organisation as a structure of business services, and analyses isolated programme management from the authority and accountability perspectives. It concludes with an explanation of the specifics of funding, return on investments (ROI) and project delivery practice relevant to service orientation.

...

4.1.2. How Business Flexibility Affects Business Efficiency

In an environment that changes fast and frequently, an organisation has to embrace change if it wants to win. A company's behaviour and even its organisation have to be efficient all the time to survive and to differentiate the company in the market. To keep organisation efficient in the changing environment it needs to be able to innovate rapidly. Its ability to do so depends on the organisation's flexibility.

With regard to changes, organisation flexibility may be measured via parameterisation. In particular, we can identify the following flexibility (FIX) parameters:

- how quickly a change may be adopted in the business solution (time-to-market) - TIM. Business solution in this context includes both Business and Technical means used within the solution
- how costly it will be to adopt a change in the business solution (actual investment) - INV
- how costly it will be to integrate the change into the existing business solution landscape (impact) - IMP.

For example, let us assume we have an Entitlement System for credit risk management. Within the process of a financial report review, we identify a need for a New Reviewer role. The T2M parameter should indicate the time spent in the organisation from the formulation of the New Reviewer role till the first person is on-boarded into the role in the Entitlement System; the INV is how much the realisation of this role costs the organisation, and the IMP is the cost of adjustment of existing user profiles, security access groups, entitlement administrative console and similar surrounding elements affected when the new role is released into the live Entitlement System.

Despite differences in the FLX parameters - time and money items - we can conclude that

$$\mathbf{max (FLX) = min \{ I (T2M+INV+IMP) \}}$$

That is, if the time-to-market for the change is short and investment into the change implementation is small but the cost of re-balancing the rest of the technical systems and business operational processes is very high, we can conclude that the organisation is not at its maximum flexibility, i.e. it may be effective but not efficient yet.

Efficiency becomes a crucial factor for the organisation when the amount of change that needs to be adopted exhausts the company's resources. Surprisingly, a common practice for making the organisation more efficient and responsive to its environment is a reduction of the number of management levels and human resources within the organisational hierarchy. Some businesses feel that 'slowing down' or downsizing in this way is the best way to weather the downturn.

In theory, this results a leaner and more flexible organisation. Often, however, it is simply a short-term cost-cutting exercise that does not bring about any fundamental change to the Business. In reality, cutting staff numbers and management levels reduces capabilities, and where there is no realistic examination of why the organisation has a deficit of efficiency, this does not necessarily present enhanced profits over the long term - it may, in fact, lead to collapse. Moreover, a loss of loyalty and competitive spirit among staff can in fact reduce flexibility as it can cause them to lose interest in the consequences of one change adaptation versus another one, i.e. integrity degrades and interdependencies grow to the point where a standstill is reached.

Assume, for example, that someone is running a horticulture business on the coast. During a

normal, calm season, the flowers are covered to protect them from the nightly breeze. If the wind is stronger, the cover should be made stronger but the business will still be able to function. However, in a period of constant storms even a strong cover may run the risk of blowing down and allowing the wind to destroy the blooms. If this takes place, the business will also be destroyed. This is what happens in an economic crisis - it is so strong and so long that just 'slowing down', even reaching a 'standstill', threatens the very existence of the business, not just its revenue. If the 'storm' affects an area where the business is expanding, it is still possible to roll back and survive in spite of some losses but it is probably better for the business to ride out the 'storm' and to prove its capabilities in new market.

Businesses that 'slow down' and mark time, or actually trim the organisation find that they harm their serviceability and service orientation, as well as their flexibility. This results in a lower level of sustainability and a higher risk of a total crash. Another reason for failure may be that the Business carries on doing the same things and in the same way but slower, in smaller volumes, and with fewer reserves. When a Business shrinks, it gradually loses the ability to analyse the market situation, new conditions, changed demand, and so on; all these things together 'kill' flexibility. As a result, the low-level processes and operations in the Business get more and more out of sync with the changing external environment, which makes the organisation more fragile rather than more efficient.

Monolithic corporate technology only adds more pain to the Business when it finds it necessary to move quicker. The fact is that business strategy and related business operations can change faster than separated software and hardware infrastructures, regardless of whether they are integrated or not. There are many contributors to this phenomenon. To name just a few of them, there can be miscommunication between business and technical semantics, tones of legacy systems with lost technical knowledge, follower role of IT without proper vision into new business trends in the company, process-oriented operational structure of IT, potential conflicts of professional interest between technical and business people, dependencies on multiple disconnected software packages and vendors.

Thus, if a Business is looking for efficiency, it has to take care of flexibility in both areas of the organisation - in the business itself and in the technology. Modifying the organisation of Technology, the major concern ought to be how easily and quickly Technology can change its automated business solutions. This relates as much to the Technology in-house as to the distant SaaS or Cloud Computing. The latter offers immediate cost reduction on technical resources but the associated business risks, including trust and flexibility, have not been fundamentally analysed and considered yet.

...

4.3.2 Funding Service-Oriented Initiatives

4.3.2.1 Funding by Business Programme

Only in Wonderland can a market exist without investments and profits. For the service-oriented business market within and outside of the organisation we need to know how to deal with investments and their returns. Business does not always need ROI when using services; sometimes

business just needs cost reduction on existing solutions. However, ROI is the major indicator of the investment efficiency for the majority of business initiatives. In Service-Oriented Enterprise, modifications and new development are usually associated with financing of business service-oriented solutions. Financing is also one of the most important factors in the transition to SOE. Many managers and CIOs claim that they are unable to proceed with SO initiatives because of insufficient investment, or because they cannot clearly demonstrate business and the financial benefits of SOS to solicit such investment.

The most important point about the financing of SOS is the understanding that a single isolated service does not provide service orientation for the company; neither do two services, or even three services. Only an integral composite approach can extract maximum business value from the service orientation. This also applies to the creation of service runtime infrastructure. As discussed later in this book, such infrastructure requires two Repository systems - one for keeping records about available services and the other for technical and business policies. The Repositories are relatively complex systems (if you wish to retain flexibility and service life-cycle management) and project-based 'as needed' funding of them does not work. Thus, the organisation has to finance service orientation at the level where the composition or combination of services makes business sense and where investments into the service infrastructure may be justified. This is a level of, at least, Business Programme within or across LOB.

Steve Jones talks about this topic briefly and comes to the same conclusion [2]. A Programme has clearer objectives and a greater orientation on the final business result than any one individual project. Managing investments inside the Programme is much more resource efficient than managing them at the project level. This is why the Programme might be the lowest level of financial surveillance in the service-oriented world. To illustrate this conclusion, assume we have to modify a business function comprising three features. Each feature is represented by a self-containing autonomous service or unit of functionality. The change affects all three services and funding is project-based. Due to the relative independence of the services, each service team has to have its own project. However, the business of one of the services has different priorities and does not want to invest adequately in this modification at a particular time. The result is a foregone conclusion - the business function cannot be modified and the organisation loses business advantage in the market. By contrast, if the funding were programme-based, i.e. given to the cross-divisional management under the cross-function authority, the conflict of priorities would be resolved by financial means without placing the business function at risk.

In addition to the programme-based funding of SOS, there are other areas where project-based funding may be still efficient. Examples are:

- a) maintenance of the run-time environment for technical tools;
- b) the function of Help Desk service that supports human collaboration technologies on desktops like e-mail as well as packaged applications such as Enterprise Resource Planning tools, etc.;
- c) the function of information record-keeping and storage of data, i.e. shared information resources for business services.

This means that IT and Business should not be split totally into service-oriented units of functionality; there should still be a solid layer of shared resources in both Business and IT. Service

orientation is a powerful model, but not a panacea for everything.

4.3.2.2 ROI for Service Usage Scenarios

For the past four or five years many analysts and organisations have tried to come up with a comprehensive calculation of ROI for SOS. The problem they faced was related to the strictly technical interpretation of 'what SOS is'. On a positive note, there is a category of projects that have succeeded in generating of ROI. For instance, so-called 'green SOA' and the associated 'green ROI' were obtained from reuse of software applications that allowed reduction in required computation power; i.e. reducing needs in terms of electricity, ventilation, heat and so on. One thing is unclear, though - why could IT not reuse its assets before with the same 'green' effect? What is service-oriented in this solution? Reuse does not constitute SOA by itself; it is one of SOA mechanisms. Actually, SOA is an architectural style of solutions; it is the SOS that can produce ROI.

Many developers suggest that there is only one approach to ROI calculation - so-called 'cost avoidance'. It is based on leveraging available services compared to otherwise new development. At a glance this seems reasonable, because the cost of service comprises only initial service development cost, plus the cost of all service changes. However, there is a limit to how many times IT can use such ROI source. To forecast the total potential ROI at a business meeting, it is possible to multiply the ROI from the first case by the number of service reuse. However, when IT approaches the second reuse, Business may ask: why do you trace ROI from an empirical avoided development when we now have the cost of real service development? And actual ROI drops far below its initial promise 'ROI for SOA' is relatively popular topic but it is still not supported by a commonly accepted ROI calculation methodic. There are two dominant groups of opinions on the ROI for services - '*there is ROI*' and '*there is no ROI*'. ZapThink has very good reflection on the first group [3]: '*ROI calculations for SOA projects can vary greatly ...In many cases, SOA implementations can provide a clear, positive ROI from the first day a Service goes live. However, it is more likely that ROI expectations, like SOA implementations, should be iterative in nature, frequently assessed, and composite!*

The opinion of the second group is illustrated by Randy Heffner of Gartner (who incidentally does not share this view): '*Any attempt to assign a specific ROI to SOA should be viewed with heavy skepticism!* Moreover, some think that '*while ROI can provide a rough picture of the value of a project, the relevance and accuracy of ROI is often suspect. For a murky concept such as SOA, the math grows even fuzzier*' [4], which is well illustrated in Figure 4.3.2.2.

It is obvious that traditional ROI calculations, with their project-based nature, are inadequate to SOA development. Indeed, if an investment is considered as per project, then there might be a

preference to construct a new patch rather than build a new service. Frequently, a new patch is cheaper and it works in the short term. But sometimes if you invest 10-15 per cent more on a new service it works for longer and returns more

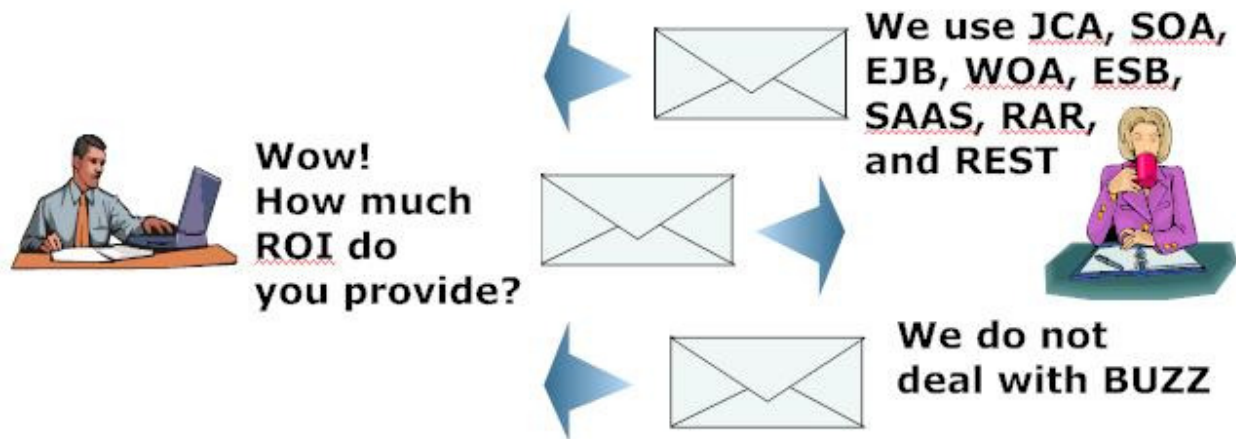


Figure 4.3.2.2. A word about ROI

than the patch does. In very many cases, investing in patches is similar to the situation where a man has lost his watch on the dark side of the street but searches for it on the light side just because he can see better over there.

David Linthicum is one of the SOA leaders who tries to define methods of determining ROI for SOA. He has identified two sources of SOA ROI: service reuse and agility [5] with following metrics:

*'Under the concept of service reuse...
 The number of services that are reusable.
 Complexity of the services.
 The degree of reuse from system to system/'*

The metrics for the first group are:

- *'degree of change over time'* is the number of times over a period time that the business reinstates itself
- *'ability to adapt to change'* is a number that points to the company's ability to adopt the change over time
- *'relative value of change'* is the revenue obtained as a direct result of change adoption.

We can discuss how the listed metrics play within the Absolute ROI and Relative ROI, as well as with

different types of services. By Absolute ROI we understand the ROI produced by the use of a newly developed service-oriented solution (SOS) while Relative ROI is a more hypothetical ROI calculated relative to the already existing solution (what ROI would be if we replace the existing solution with SOS).

Elaborating on the service reuse, we note that the metric *'number of services that are reusable'* does not lead to any ROI by itself but can help to find whether the investment has any ROI potentials for the given period of time (time-window). If the ratio of the *'degree of reuse'* metric to the *'number of services that are reusable'* metric is less than 1, then a part of the investment does not have a ROI potential for the time-window under consideration.

The metric *"Complexity of the services"* - average number of business functions per service - is rather ambiguous because a) it may not reflect anything if all services are of one function only; b) it can mislead in ROI calculations for tactical versus strategic investments. For example, the tactical services may have a very small number of functions while the strategic ones may have far more; an average number of functions used in tactical calculations can increase ROI based on this metric and decrease it for the strategic cases. This may result in a decision to create more tactical services, which is an obvious mistake from the ROI perspective.

Talking about ROI for service reuse 'as is', we can identify two ROI contributors:

- 1) reuse of existing service instead of building another one (Relative ROI)
- 2) service creation comparative to the baseline of non-service-oriented solution (Absolute ROI)

We need to note that the first contributor has a hidden dependency on the time-frame that the ROI is calculated for: if it is a short time, weeks or months, the ROI may be positive. If it is calculated in quarters or years, the ROI may become quite negative due to the added high cost of maintenance/modifications for the widely 'as is' reused services.

Service reuse 'by extension' has one additional contributor:

- 3) the number of extensions comparatively to the first service version (Relative ROI)

This type of reuse includes variations of the message namespaces used to adopt the existing service to new business needs, i.e. to the changes. The reuse 'by extension' encounters much less dependency on the time factor than the reuse 'as is' and can provide for more stable ROI for tactical and strategic services.

We do not claim a new methodology of ROI for SOS but just point out the direction of ROI calculations. First, we describe the cost of SOS and its reflection in different Absolute ROI and Relative ROI. So, the SOS cost may be expressed as follows:

SOS COST = {READ IN THE BOOK}

where SOS Development Cost includes development cost of all projects contributed into SOS; Maintenance Cost includes maintenance cost of all elements of SOS for given time-window; Cost of Fixed Payments includes all fixed payments like licences and fees caused by all elements of SOS for given time-window. The (SOS COST + SOS CHANGE COST) constitutes the cost of SOS ownership for given time-window.

Then, we need to identify a cost of baseline solution, i.e. the cost of non-SOS that might be encountered in absence of SOS:

BASELINE COST = {READ IN THE BOOK}

Next, we define a time factor: if we can calculate SOS ROI for given time-window and assume that conditions do not change for a prospective period of time, we can find how many times the time-window related ROI may be repeated for the prospected period of time. Thus, the time factor may be like this:

TIME-FACTOR = {READ IN THE BOOK}

The Absolute ROI may be generated when a standard-based SOS, e.g., service or service intermediary, is used. The foundation of the ROI may be, for example, an open-source implementation of the SOS, which can replace a proprietary/vendor solution and reduce/eliminate license cost. Another basis of the Absolute ROI links to the Relative ROI from reuse -one service can fulfill a business function or feature that, when reused, allows for the removal of redundant systems/applications and a corresponding reduction in maintenance costs.

Now, we can review the ROI for SOS for four cases: direct replacement of existing solution, redundancy minimisation, service creation comparatively to the baseline, and service reuse for given time-window:

SOS REPLACEMENT ROI = {READ IN THE BOOK}

SOS REDUNDANCY MINIMISATION ROI = {READ IN THE BOOK}

SOS CREATION To BASELINE ROI = {READ IN THE BOOK}

SOS REUSE ROI = {READ IN THE BOOK}

The SOS Change Cost includes the cost of change for entire SOS, for all the elements, if affected. My researches have shown [6] that the SOS Change Cost and Number Of Changes are usually lower for the reuse 'by extension' than for the reuse 'as-is'.

Finally, the agility metrics that David Linthicum has specified represent higher level of abstraction than reuse metrics and we have to deal with them differently. These metrics have a flavour of hypothetical possibilities. For example, what a service ROI might be calculated by using "*degree of change over time*" metric if the service had not been used to implement the changes? The same relates to two other metrics.

Nevertheless, it appears there may be a certain integral formula incorporating agility metrics into

SOS ROI. Principally, we might need to find how many times the same SOS was used for particular business change happened for given time-window (Number Times SOS used), and, then, how much ROI can be produced as a direct result of using SOS for this business change. I would name this a Change Agile Service-Oriented Solution (CASOS) ROI:

$$\text{CASOS ROI} = \frac{\text{number of business changes happened}}{(\sum\{\text{READ IN THE BOOK}\}), \text{ per business change}}$$

where SOS ROI is an accumulation of all types and cases of ROI for particular SOS. While CASOS ROI is hypothetical as well It gives the quantifiable perspective of SOS efficiency and practical business reasons to go with SO solution, or not.

In the SOE, we do not need to talk about ROI for SOA *per se* because an architecture style and architecture are only organising elements of the solution for the business problem; it is the SOS which may result in ROI. ROI is not rocket-science and if you know exactly what you are doing in each SO initiative, the Project Manager or Architect can easily do the ROI calculation.

In this section, we discussed the service-related part of SOS ROI, but this is not the only source of ROI. Additionally, SOS ROI can include a contribution from a Rules Engine used instead of hard-coded logic in the decision points of the business process implementation. Other ROI contribution may come from the reuse of execution context policies. For such cases, service orientation is only an environment allowing policy reuse or policy modification as a means of agility to market demands. All these contributors are the reason why I use a notion of service-oriented solution - SOS - instead of a notion of service in the formulas. I am sure people will find other ROI metrics to consider for different cases.

...

4.4 Service-Oriented Business Organisation

A Service-Oriented Enterprise is by definition an SO Business Organisation. SOE is not about technology or IT, it is about the organisation of an enterprise along service-oriented principles. Service orientation, to provide maximum benefits, has to be adapted by the business part of the organisation and cascaded into its technical part. If SO development happens in IT only, its business value is limited because it does not provide the organisation with real SOS for market needs.

To become service-oriented, i.e. to become flexible and efficient, the organisation has to re-shape itself as shown in Figure 4.1.3. The shift to SOE changes the concept of the enterprise organisation, which is illustrated in Figure 4.4. First, the business has to adapt its outlook so it is driven by business function and cross-functions, then it has to dissolve the barrier between Business and Technology, and then it has to utilise its internal self-containing autonomous services, i.e. units of functionality, and their combinations to resolve external tasks.

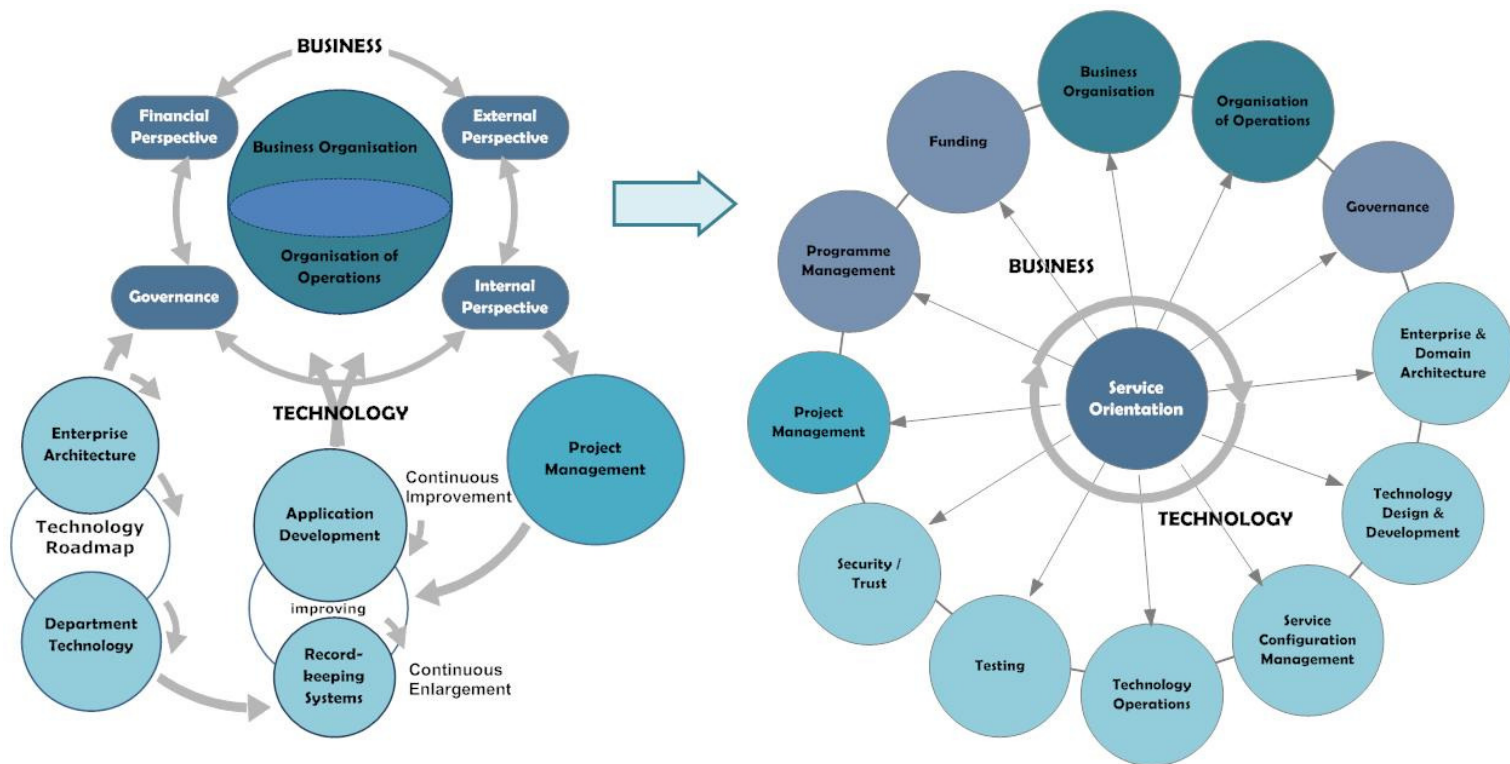


Figure 4.4. Influence of service orientation on the concept of the enterprise organisation

Many warn against 'boiling the ocean' during service-oriented innovations; i.e. trying to do something that's so ambitious it's effectively impossible. This is wise advice; it is impossible to change relationships, processes, technology and people simultaneously even if you are building an organisation from 'scratch'. Figure 4.4 shows that practically all important organisational functions are affected when service orientation is finally realised. To achieve such a cohesive result, an organisation has to undergo a long process of change. The Yule of thumb' in this process is that every change may be quite narrowed in scope but implemented end-to-end, i.e. all dependent aspects should be addressed together, within the single Programme and compact time-window.

In this book, I discuss some fundamental statements derived from SO principles when they are applied to well-known aspects of the relationship between Business and Technology and within Technology itself. In doing so, I am trying to establish WHAT is WHAT and WHY, and only then to move into HOW. You will find discussions and some practical conclusions and recommendations on how to do 'regular' things in the SO manner, how to recognise the steps to SOE and how to walk some of the steps yourself. The following chapters are quite technical, but their purpose is to rescue Technology from the consequences of technical-only SOA and create the basis for proper SO collaboration with Business.